AERONAUTICAL SYSTEMS CENTER
MAJOR SHARED RESOURCE CENTER

ARCHIVAL SERVER
USER'S GUIDE

1 October 2005

Release Notes

Issued 8 April 1997.

Initial release.

Issued 15 August 1997.

Update to directory structure and modified cover.

Issued 11 December 1997.

Format changes made to standardize all user guides.

Issued 15 November 1999.

Rewritten to reflect changes due to StorageTek ASM and SAM-QFS.

Issued 2 November 2001.

Removed references to IBM SP P2 (hpc02), archive command wrapper.

Added references to Compaq GS320 (hpc08).

Issued 7 December 2001.

Fixed minor C shell errors in examples

Issued 30 January 2002.

Corrected machine names

Issued 1 April 2002.

Removed references to SGI Origin 2000 (hpc03)

Issued 1 June 2002.

Added references to Compaq ES45 (hpc-09, hpc-10)

Issued 1 October 2002.

Added references to archive (PSToolkit), safeftp

Issued 30 May 2003.

Added references to SGI Origin 3900 (hpc11)

Issued 1 October 2003.

Removed references to Compaq GS-320 (hpc08)

Issued 1 October 2004

Removed references to IBM P3 (hpc04)

Updated server type to SunFire 6800

Issued 1 October 2005

Removed references to Compaq ES-40 (hpc05)

Added references to SGI Altix (eagle)

Table of Contents

1 Introduction..1

2 Using the Archival Server 2

2.1 Creating Subdirectories...2

2.2 Using Archive Files 3

2.3 Methods for Storing and Retrieving Data..3

2.3.1 Using *cp*/*mv* to Store Files and/or Directories 4

2.3.2 Using *cp* to Retrieve Migrated Files and/or Directories 4

2.3.3 Non-Kerberized rcp on the ASC MSRC Application Servers.........................5

2.3.4 Using *rcp* to Store Files and/or Directories 5

2.3.5 Using *rcp* to Retrieve Files and/or Directories 6

2.3.6 Using *ftp* for Storing Files and/or Directories...6

2.3.7 Using *ftp* for Retrieving Stored Files and/or Directories 7

2.3.8 Using *archive* to Store Files and/or Directories...8

2.3.9 Using *archive* to Retrieve Files and/or Directories...8

2.4 Other Commands 9

2.4.1 Non-Kerberized rsh on the ASC MSRC Application Servers.........................9

2.4.2 Determining the Status of Files on the Archival Server 9

2.5 Machine names 10

A Quick Reference 11

A.1 Creating Subdirectories on the Archival Server 11

A.2 Saving Files to the Archival Server 11

A.3 Retrieving Files from the Archival Server...12

A.4 Non-Kerberized rcp on the ASC MSRC Application Servers..................................13

A.5 Non-Kerberized rsh on the ASC MSRC Application Servers...................................13

B Tools 14

B.1 cp...14

B.2 mv 14

B.3 rcp 15

B.4 scp 15

B.5 ftp..15

B.6 archive..16

B.7 safeftp...16

B.8 tar 16

B.9 gzip..17

B.10 gunzip...17

B.11 gzcat 17

B.12 sls 18

1. Introduction

This document provides an overview of the archival capabilities of the Aeronautical Systems Center (ASC) Major Shared Resource Center (MSRC) and how they may be used. The ASC MSRC is physically located in Building 676 of Area B, at Wright-Patterson Air Force Base, near Dayton, Ohio.

The archival storage system for the ASC MSRC (hence, denoted as the archival server) is currently configured with a total capacity of 500 terabytes of near-on-line storage. The system consists of a set of dual Sun SunFire 6800 cluster servers with on-line RAID storage attached to dual-combined Storage Technology Corporation (StorageTek) 9310 tape silos. This facility provides long-term storage for user files and data backup files.

The archival server uses the StorageTek Application Storage Manager (ASM) to move data files to tape when the files are inactive. The ASM is also used to move the files back to disk when a user needs to access the contents of the files. The file status information of the files that are migrated to tape remains on disk, which gives the appearance that the files are still stored on disk.

The archival server is available from the ASC MSRC High Performance Application Servers and the Scientific Visualization workstations.

Section 2 provides explanations of the various methods for storing and retrieving files and directories on the archival server, and contains recommendations of appropriate methods. Appendix A is a quick reference to using the archival server. Appendix B discusses useful commands with the options that are relevant to the archival process.

2. Using the Archival Server

Users are not allowed to login into the archival server, but they are allowed to connect via the `ftp`, `rcp`, `rsh/remsh` (remote shell), and `ssh/scp` (secure shell/copy) commands. In addition, the locally installed utilities `archive` and `safeftp` provide other interfaces to the archival server. Each user is given an *archive directory,*[1] the user’s archive directory, on the archival server which is network file system (NFS***)***-mounted on the ASC MSRC Application Servers. Thus, users can directly access their archive directory from any ASC MSRC Application Server. Note: as of the introduction of the SGI Altix, eagle, the NFS-mounted archive directories are *only* available on the interactive front-end nodes.

The ASC MSRC startup scripts create the environment variables **ARC** and **msas** associated with the archival server. ***ARC*** contains the path of the user’s archive directory. The user's archive directory path can be simply written as **$ARC** as used in the examples in this User’s Guide. ***msas*** contains the hostname of the server on which the user’s archive directory is mounted. **$msas** is used where a machine hostname is required in this User’s Guide. It is recommended that users use **$msas** and **$ARC** for the archive server name and archive directory to avoid problems when these values change.

The archival server uses the StorageTek ASM to maintain the near-on-line storage system. ASM periodically migrates data files to tape, but retains the directory information about the files on disk. When a user attempts to access the data in a migrated file (*e.g.*, attempting to copy the file), ASM will automatically copy the data file back to disk.

The archival server provides large file-space capacity at the expense of longer access times (*i.e.*, the time needed to read a file from tape) than from disk. Since the access time on this system can be much longer than the access time on other file systems and file servers, the archival server is best suited for long-term and permanent storage and the storage of large files. Infrequently accessed files, large data files, and file archives of entire directories are best stored on the archival server.

Users should not directly use files located on the archival server. Changes made by editing or appending to these files may not be saved if one of the Sun servers fails. Users should copy the files from the archival server to the system where the files are needed and copy the modified files back to the archival server. Possible destinations are the user's **$HOME** directory, the global `/scratch` directory, the `/workspace` directory of an appropriate ASC MSRC High Performance Server, or space on a local disk (such as a hard disk attached to the user's local workstation).

NOTE: The ASC MSRC Archival Storage System is not a replacement for the normal backups made by the system administrators and should not be used as an extension of a user's disk allocation.

2.1 Creating Subdirectories

Subdirectories are used to organize files within the user's archive directory. In the examples below, files are copied to and from subdirectories that must already exist.

[1] The user's archive directory is the user's home directory on the archival server.

To create a new subdirectory, use the **mkdir** command:

```
cd $ARC
mkdir dirname    or  mkdir $ARC/dirname
```

To create a ***link*** to the new directory in the user's ***current working directory***, use the **ln** command:

```
ln -s $ARC/dirname ./dirname
```

where **$ARC/**`dirname` is the new subdirectory on the archival server. The user can access the subdirectory by referencing the link `dirname`.

Users can also put their data files directly in their **$ARC** directory and create no subdirectories. This arrangement may become unwieldy as more files are stored in the user's archive directory.

2.2 Using Archive Files

Although **rcp**, **scp** and **cp** can transfer entire directories to and from the archival server, users are strongly advised to use archive files instead because they are much more efficient for storage and transfer. The **tar** (tape archive) command creates and extracts from archive files. **tar** archive files, which store the directory structure as well as the file contents, are recommended for storing directories and collections of related files.

- Creating an archive file using **tar**:

```
tar cf tarfile file-1 ... file-N dir-1 ... dir-N
```

where `tarfile` is the archive file created by **tar**, and `file-1 ... file-N` and `dir-1 ... dir-N` are the files and directories to be archived.

- Extracting from the archive file

```
tar xf tarfile
```

where `tarfile` is the name of the archive file.

2.3 Methods for Storing and Retrieving Data

The user has three methods for copying files to and from the archival server:

- The **cp** command on an ***NFS***-mounted filesystem: this method is recoverable when a server fails, but will be slower than the other methods.. It can also be used in shell scripts.
- The **rcp** and **scp** commands: this method is more complicated to use than **cp**. Only non-Kerberized **rcp** (see Section 2.3.3) can be used in batch scripts. Only Kerberized **rcp** and **scp**can be used to transfer files between the archival server and machines outside the ASC MSRC. Since **rcp** and **scp** have the same syntax, only **rcp** is used in the examples.
- The **ftp** command: **ftp** commands cannot be incorporated into shell scripts due to security restrictions. Only Kerberized **ftp** can be used to transfer files to and from the archival server.

- The **archive** command: the **archive** command is a locally installed utility that provides a uniform interface to the archival servers at DoD MSRCs. The **archive** command can be used from within batch scripts.
- The **safeftp** command: the **safeftp** command is a locally installed utility that provides an interface similar to **rcp** and **scp**, but using **ftp**, which allows for some error checking and restart capabilities. Since it uses **ftp**, it cannot be used in batch scripts. Since the form is similar to **rcp** and **scp**, it is not described in the following examples.

NOTE: In the examples below, the files are copied to and from the user's ***current working directory*** for convenience. This is not a requirement. Also, since **rcp**, **scp**, **ftp** and **rsh**/**remsh** begin in the user's ***archive home directory*** **$ARC** on the archival server, the relative path `dirname` is equivalent to **`$ARC/`**`dirname`.

2.3.1 Using *cp*/*mv* to Store Files and/or Directories

- To store file(s)

```
cp file-1 ... file-N          or   mv file-1 ... file-N
  $ARC/dirname                       $ARC/dirname
```

where `file-1 ... file-N` are the files to be stored, and `$ARC/dirname` is the destination directory on the archival server.

- To store directories or projects

1. Create an archive file `tarfile` using **tar:**

```
tar cf tarfile  file-1 ... file-N dir-1 ... dir-N
```

where `tarfile` is the archive file created by **tar**, and `file-1 ... file-N` and `dir-1 ... dir-N` are the files and directories to be archived.

2. Store the archive file `tarfile`:

```
cp tarfile
$ARC/dirname                or   mv tarfile $ARC/dirname

rm tarfile
```

where **`$ARC/`**`dirname` is the destination directory on the archival server.

2.3.2 Using *cp* to Retrieve Migrated Files and/or Directories

- To retrieve file(s)

```
cp $ARC/dirname/file-1 ... $ARC/dirname/file-N  .
```

where **`$ARC/`**`dirname` is the source directory on the archival server, `file-1 ... file-N` are the files to be retrieved, and the last '**`.`**' means the current directory.

- To retrieve directories or projects

1. Retrieve the archive file `tarfile`:

```
cp $ARC/dirname/tarfile .
```

where `$ARC/dirname` is the source directory on the archival server, and the last ‘`.`’ means the current directory.

2. Extract from the archive file and remove it:

```
tar xf tarfile
rm tarfile
```

2.3.3 Non-Kerberized *rcp* on the ASC MSRC Application Servers

Because users cannot get a Kerberos ticket for batch jobs, they must use the non-Kerberized **rcp** in batch scripts. The following table lists the full pathnames for non-Kerberized **rcp** on the ASC MSRC Application Servers.

hpc09, hpc10	/usr/bin/rcp
hpc11	/usr/bsd/rcp
eagle	/usr/bin/rcp

2.3.4 Using *rcp* to Store Files and/or Directories

- Storing file(s)

```
rcp file-1 ... file-N ${msas}:dirname
```

where `file-1 ... file-N` are the files to be archived, and `dirname` is the destination directory on the archival server.

- Storing directories or projects

1. Create an archive file using **tar**

```
tar cf tarfile file-1 ... file-N dir-1 ... dir-N
```

where `tarfile` is the archive file created by **tar**, and `file-1 ... file-N` and `dir-1 ... dir-N` are the files and directories to be archived.

2. Store archive file and delete local copy

```
rcp tarfile ${msas}:dirname
rm tarfile
```

where `dirname` is the destination directory on the archival server.

2.3.5 Using *rcp* to Retrieve Files and/or Directories

- Retrieving file(s)

```
rcp ${msas}:dirname/file-1 ...
${msas}:dirname/file-N .
```

where `dirname` is the source directory on the archival server, `file-1 ... file-N` are the files to be retrieved, and the last '`.`' means the current directory.

- Retrieving directories

1. Retrieve archive file `tarfile`

```
 rcp ${msas}:dirname/tarfile .
```

where `dirname` is the name of the source directory on the archival server, and the last '**.**' means the current directory.

2. Extract from the archive file `tarfile` and remove it

```
tar xf tarfile
rm tarfile
```

2.3.6 Using *ftp* for Storing Files and/or Directories

- Storing file(s)

1. Open an **ftp** session to the archival server:

```
ftp $msas
```

2. At the prompt press the return key to accept the default user name **username**:

```
Name (msasX:username):<CR>
```

3. Change to the destination directory `dirname` on the archival server

```
cd dirname
```

4. Put the file(s) to the archival server:

```
 put file-1  or  mput file-1 ... file-N
```

where `file-1 ... file-N` are the files to be archived.

5. End the **ftp** session:

```
quit
```

- Storing directories or projects

1. Create an archive file using **tar**:

```
tar cf tarfile file-1 ... file-N dir-1 ... dir-N
```

where `tarfile` is the archive file created by **tar**, and `file-1 ... file-N` and `dir-1 ... dir-N` are the files and directories to be archived.

2. Open an **ftp** session to the archival server:

```
ftp $msas
```

3. At the prompt press the return key to accept the default user name **username**:

```
Name (msasX:username):<CR>
```

4. Change to the destination directory `dirname` on the archival server:

```
cd dirname
```

5. Store the archive file `tarfile` on the archival server:

```
put tarfile
```

6. End the **ftp** session:

```
quit
```

7. Delete the local copy of the archive file `tarfile`:

```
rm tarfile
```

2.3.7 Using *ftp* for Retrieving Stored Files and/or Directories

- Retrieving file(s)

1. Open an **ftp** session to the archival server:

```
ftp $msas
```

2. At the prompt press the return key to accept the default user name **username**:

```
Name (msasX:username):<CR>
```

3. Change to the source directory `dirname` on the archival server:

```
cd dirname
```

4. Retrieve the file(s) from the archival server:

```
 get file-1  or  mget file-1 ... file-N
```

where `file-1 ... file-N` are the files to be retrieved.

5. End the **ftp** session:

```
quit
```

- Retrieving directories

1. Open an **ftp** session to the archival server:

```
ftp $msas
```

2. At the prompt press the return key to accept the default user name **username**:

```
Name (msasX:username):<CR>
```

3. Change to the source directory `dirname` on the archival server:

```
cd dirname
```

4. Retrieve the archive file `tarfile` from the archival server:

```
get tarfile
```

5. End the **ftp** session:

```
quit
```

6. Extract from the archive file `tarfile` and remove it:

```
tar xf tarfile
rm tarfile
```

2.3.8 Using *archive* to Store Files and/or Directories

- Storing file(s)

```
archive put -C dirname file-1 ... file-N
```

where `file-1 ... file-N` are the files to be archived, and `dirname` is the destination directory on the archival server.

- Storing directories or projects

1. Create an archive file using **tar**

```
tar cf tarfile file-1 ... file-N dir-1 ... dir-N
```

where `tarfile` is the archive file created by **tar**, and `file-1 ... file-N` and `dir-1 ... dir-N` are the files and directories to be archived.

2. Store archive file and delete local copy

```
archive put -C dirname tarfile
rm tarfile
```

where `dirname` is the destination directory on the archival server.

The **archive** command can also create archive files using **tar** implicitly. See the man pages for more details.

2.3.9 Using *archive* to Retrieve Files and/or Directories

- Retrieving file(s)

```
archive get -C dirname file-1 ... file-N
```

where `dirname` is the source directory on the archival server, `file-1 ... file-N` are the files to be retrieved, and the last '`.`' means the current directory.

- Retrieving directories

1. Retrieve archive file `tarfile`

```
archive get -C dirname tarfile
```

where `dirname` is the name of the source directory on the archival server.

2. Extract from the archive file `tarfile` and remove it

```
tar xf tarfile
rm tarfile
```

The **archive** command can also extract from archive files using **tar** implicitly. See the man pages for more details.

2.4 Other Commands

The **sls** command is an extended version of the Unix **ls** command. These commands are available via **rsh** or **ssh** to the Sun servers. The command line used for these commands should be in the form:

```
rsh|ssh $msas 'command options filelist'
```

Only non-Kerberized **rsh** (see Section 2.4.1) can be used in batch scripts. Only Kerberized **rsh** or **ssh** can be used to connect to the archival server from machines outside the ASC MSRC.

As an alternative, a wrapper for **sls** exists on the ASC MSRC Application Servers. This wrapper starts a remote shell on the archival server and passes any arguments entered to the indicated command. For example: **sls -l** is equivalent to **rsh $msas** '**sls -l**'.

NOTE: when using these wrapper utilities, arguments that contain wildcards (*e.g.*, *, ?, [,]) need to be quoted to avoid being expanded by the shell on the local machine.

2.4.1 Non-Kerberized *rsh* on the ASC MSRC Application Servers

Because users cannot get a Kerberos ticket for batch jobs, they must use the non-Kerberized **rsh** in batch scripts. The following table lists the full pathnames for non-Kerberized **rsh** on the ASC MSRC Application Servers.

hpc09, hpc10	/usr/bin/rsh
hpc11	/usr/bsd/rsh
eagle	/usr/bin/rsh

2.4.2 Determining the Status of Files on the Archival Server

The **sls** command is an extension of the Unix directory-listing command **ls**.

- Standard long listing

Typing

`ls -l $ARC/test.dat` (NFS mounted)

or

`sls -l $ARC/test.dat`

produces

```
-rw-------    1 user user 1000 Sep 10 12:50 /msas031/user/test.dat
```

- Two-line listing

  Typing

  ```
  sls -2 $ARC/test.dat
  ```

  produces

```
-rw-------   1 user user  1000 Sep 10 12:50 /msas031/user/test.dat
O-a------- ---  sg
```

In addition to the output of the standard long listing, the ‘‘O’’ in the second line indicates that the file is off-line, and must be staged before being accessed.

- Detailed listing

  Typing

  ```
  sls -D $ARC/test.dat
  ```

  produces

```
/msas031/user/test.dat   mode: -rw-------  links:   1  owner: user  group: user
length:      1000  inode:      608
offline;  archdone;
copy 1: ---- Sep 10 12:51     27eb4.1    sg 000305
access:        Sep 10 12:50  modification: Sep 10 12:50
changed:     Sep 10 12:50  attributes:      Sep 10 13:52
creation:      Sep 10 12:50  residence:    Sep 10 13:52
```

Refer to the on-line **man** pages for more information regarding **sls**.

2.5 Machine names

$msas contains the name of the machine where the user’s files reside.

A. Quick Reference

These instructions assume that the reader has some familiarity with UNIX and a working knowledge of the ASC MSRC environment. For explanations and more details, see Section 2 and the Appendix B.

A.1 Creating Subdirectories on the Archival Server

These steps are optional; you may place your archives directly into your archival storage area. The link in the current directory is created for your convenience.

```
mkdir $ARC/dirname
ln -s $ARC/dirname ./dirname
```

A.2 Saving Files to the Archival Server

The **tar** method is recommended for multiple files or directories.

- Method 1 -- storing an archive file

1. Create archive file:

```
tar cf tarfile file-1 ... file-N dir-1 ... dir-N
```

2. Copy archive file to archival server:

```
cp tarfile $ARC/dirname
```

or

```
rcp tarfile ${msas}:dirname
```

or

```
ftp $msas
Name (msasX:username):<CR>
cd dirname
put tarfile
quit
```

or

```
archive put -C dirname tarfile
```

- Method 2 -- storing a series of files

```
cp file-1 ... file-N $ARC/dirname
```

or

```
     rcp file-1 ... file-N ${msas}:dirname
```

or

```
     ftp $msas
     Name (msasX:username):<CR>
     cd dirname
     mput file-1 ... file-N
     quit
```

or

```
     archive put -C dirname file-1 ... file-N
```

A.3 Retrieving Files from the Archival Server

When retrieving files, the current working directory (local directory on the ASC MSRC) is the directory to which files are copied.

- Method 1 -- Retrieving an archive file

  1. Copy archive file:

```
cp $ARC/dirname/tarfile .
```

or

```
rcp ${msas}:dirname/tarfile .
```

or

```
ftp $msas
Name (msasX:username):<CR>
cd dirname
get tarfile
quit
```

or

```
archive get -C dirname tarfile
```

  2. Extract files for archive file:

```
     tar xf tarfile
```

- Method 2 -- Retrieving a series of files

```
cp $ARC/dirname/file-1 ... $ARC/dirname/file-N .
```

or

```
rcp ${msas}:dirname/file-1 ... ${msas}:dirname/file-N .
```

or

```
ftp $msas
Name (msasX:username):<CR>
cd dirname
mget file-1 ... file-N
quit
```

or

```
archive get -C dirname file-1 ... file-N
```

A.4 Non-Kerberized *rcp* on the ASC MSRC Application Servers

hpc09, hpc10	/usr/bin/rcp
hpc11	/usr/bsd/rcp
eagle	/usr/bin/rcp

A.5 Non-Kerberized *rsh* on the ASC MSRC Application Servers

hpc09, hpc10	/usr/bin/rsh
hpc11	/usr/bsd/rsh
eagle	/usr/bin/rsh

B. Tools

Among the commands used in archival storage are: **cp** (copy), **mv** (move), **rcp** (remote copy), **scp** (Secure Shell copy), **ftp** (file transfer program), **archive** (archival server copy program), **safeftp** (**rcp**-like **ftp** interface utility), **tar** (tape archive), **gzip** (file compression), **gunzip** (file expansion), **gzcat** (lists a compressed file), and **sls** (directory listing on archival server). This section summarizes the usage of these commands as they would be used with the archival server. More complete information for these commands can be found by using the UNIX **man** command on the local system.

B.1 cp

Copies files on the local machine

Usage:	**cp [-fi]** filename-1 filename-2
	cp [-fi] filename-1 ... filename-N directory

Option	Modification
-f	Overwrites existing file of the same name in the destination directory
-i	Inquires before overwriting existing file of the same name in the destination directory

B.2 mv

Moves files on the local machine. A side effect of **mv** is to allow the renaming of a file or directory.

Usage:	**mv [-fi]** filename-1 filename-2
	mv [-fi] filename-1 ... filename-N directory

Option	Modification
-f	Overwrites existing file of the same name in the destination directory
-i	Inquires before overwriting existing file of the same name in the destination directory

B.3 rcp

Copies files between machines

Usage:	**rcp** filename-1 filename-2 **rcp** filename-1 ... filename-N directory

Each filename or directory argument is either a remote name of the form `hostname:path` or a local file name. If path is a relative path, it is relative to the user's **$HOME** directory on `hostname`. If neither source or destination arguments are remote names, **rcp** reverts to **cp**.

NOTE: Kerberized **rcp** must be used to transfer files between the ASC MSRC machines and users' local machines.

B.4 scp

Copies files between machines

Usage:	**scp** filename-1 filename-2 **scp** filename-1 ... filename-N directory

Each filename or directory argument is either a remote name of the form `hostname:path` or a local file name. If path is a relative path, it is relative to the user's **$HOME** directory on `hostname`. If neither source or destination arguments are remote names, **scp** reverts to **cp**.

B.5 ftp

Transfers files between networked computer systems.

Usage:	**ftp** [**-giv**] [hostname]

Option	Modification
-g	Disables expansion of filenames containing wildcards
-i	Turns off interactive prompting during multiple file transfers
-v	Shows all responses from the remote server, as well as report on data transfer statistics

The `hostname` is either the name or IP address of the machine which the user wants to access.

NOTE: Kerberized **ftp** must be used to transfer file in the ASC MSRC environment.

B.6 archive

Copies files to/from the archival server

Usage:	**archive get -C** directory filename-1 ... filename-N **archive put -C** directory filename-1 ... filename-N

The directory argument is only necessary when the source/destination on the archival server is not the user's home directory

B.7 safeftp

rcp-like utility that uses ftp

Usage:	**safeftp** filename-1 filename-2 **safeftp** filename-1 ... filename-N directory

Each filename or directory argument is either a remote name of the form `hostname:path` or a local file name. If path is a relative path, it is relative to the user's **$HOME** directory on `hostname`.

B.8 tar

Archives and extracts multiple files and directories using a single file archive

Usage:	**tar c | r | t | x f** tarfile filename-1 ... filename-N directory-1 ... directory-N

Option	Operation
c	Creates a new archive file, then writes to it
r	Writes the named files to the end of the named archive file
t	Lists the table of contents of the archive file
x	Extracts the named files from the archive, or all the files if no filename arguments exist
f	Specifies the name of the archive file to be accessed. If the archive filename is '-', **tar** reads from the standard input or writes to the standard output. This option ***must*** be specified.

When writing the archive file, a directory name argument refers recursively to the files and subdirectories of that directory. However, when extracting from the archive file, the entire pathname of the file as written in the archive file must be specified.

B.9 gzip

Reduces the size of the named files using Lempel-Ziv coding

Usage: **gzip** [**-cf**] [filename-1 ... filename-N]

Option	Modification
-c	Writes to the standard output; no files are changed
-f	Forces compression, even if the file does not actually shrink, or the corresponding `.gz` file already exists

Whenever possible, each file is replaced by one with the extension .gz, while keeping the same ownership modes, as well as access and modification times. If no file(s) are specified the standard input is compressed to the standard output. If the file(s) already have a `.gz` suffix, **gzip** assumes they have already been compressed.

B.10 gunzip

Restore compressed file(s)

Usage: **gunzip** [**-c**] [filename-1 ... filename-N]

Option	Modification
-c	Writes to the standard output; no files are changed
-f	Forces uncompression even if the corresponding file already exists

Requires the file(s) to have a `.gz` suffix; otherwise **gunzip** assumes they have not been compressed.

B.11 gzcat

Lists contents of a compressed file

Usage: **gzcat** [**-f**] [filename-1 ... filename-N]

Option	Modification
-f	Forces uncompression even if the corresponding file already exists

Writes uncompressed output to the standard output, but leaves the compressed `.gz` file intact. This command works exactly like **gunzip -c**.

B.12 sls

List contents of directories

Usage: **sls** [**-D2**] [path-1 ... path-N]

sls is an extended version of **ls**. The following are options to display information specific to the archival server.

Option	Modification
-D	Lists a detailed description for each file
-2	Lists in two-line long format. Similar to **-l** option, but also displays file status information on the archival server.